Encoding ECMP/UCMP information in PCEP

M. Koldychev – Cisco Systems (mkoldych@cisco.com) – Presenter
M. Sivabalan – Cisco Systems (msiva@cisco.com)

With guidance from –
D. Dhody – Huawei Technologies (dhruv.ietf@gmail.com)
We are motivated by the need to encode multiple Segment Lists in SR-TE Policies, but our results are meant to be generic.

Terminology:
• ERO == Explicit Route Object
• LSP == PCEP LSP (identified by LSP-IDENTIFIERS)
Motivation

In SR-TE Policy, the optimization problem is defined by optimization objectives and constraints and the solution to the optimization problem is a set of Segment Lists.


A dynamic candidate path expresses an optimization objective and a set of constraints. The headend (potentially with the help of a PCE) computes the solution Segment-List (or set of Segment-Lists) that solves the optimization problem.

If a candidate path is associated with a set of Segment-Lists, each Segment-List is associated with a weight for weighted load balancing (refer Section 2.11 for details). The default weight is 1.
Motivation (cont’d)

Two use-cases:
1. One optimization objective yields multiple ECMP paths.
2. Multiple optimization objectives sharing ECMP.

These two use-cases are orthogonal, they exist independently of each other.

For example:
• LSP 1 and LSP 2 have different optimization objectives and ECMP is desired
• LSP 1 gets a single path X
• LSP 2 gets two ECMP paths: Y and Z
• As a result 50% of traffic goes on path X, 25% on path Y and 25% on path Z.
Use Case 1: single optimization objective yields multiple paths.

- Extend PCEP to specify the maximum number of ECMP paths that the PCC can handle.
- Extend PCEP to allow for multiple ERO objects within a single LSP, some options are possible:
  A. define a new object to “separate” the EROs and carry per-ERO attributes
  B. follow RFC 8623 for encoding multiple EROs
  C. other proposals (to be discussed on mailing list)
Analysis of Option A

We can replace a single ERO object by multiple ERO objects, separated by a new ERO-ATTRIBUTES object.

Current BNF:
<intended-path> = <ERO>

Proposed BNF:
<intended-path> = <ero-list>
<ero-list> = [<ERO-ATTRIBUTES>]<ERO>[<ero-list>]
**Analysis of Option A (cont’d)**

Define ERO-ATTRIBUTES object to carry some per-ERO attributes:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |

```
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Flags                       | Oper|
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Weight                            |
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                         Optional TLVs                         ~
+---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Support for ERO-ATTRIBUTES can be negotiated in the OPEN message, thus guaranteeing backward compatibility.
Analysis of Option B


RFC 8623 already defines a way to carry multiple ERO/RRO objects, by using a special type of END-POINTS object and an S2LS object. We can define a new END-POINTS and S2LS object types for P2P load-balancing and then we can follow the same encoding format as RFC 8623.
Analysis of Option B (cont’d)

S2LS format is almost the same as ERO-ATTRIBUTES, except that it’s missing the Weight field:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Flags                       | Oper|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Optional TLVs                   ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The weight can be either carried in an optional TLV, or it can be embedded directly as part of the new S2LS object type.

New END-POINTS object type would need to be defined, to specify that the S2LS objects that follow it are for ECMP/UCMP.
For example, suppose the PCE was computing a path from Source A to Destination X and the result was 2 EROs: \{A,B,X\} and \{A,C,X\}. Suppose that the 2 EROs have UCMP weights 2 and 3 respectively. Then the PCE would encode this as follows:

Common Header

LSP

END-POINTS (SRC=A, DEST=X)

S2LS (O=UP, WEIGHT=2)

ERO1={A,B,X}

END-POINTS (SRC=A, DEST=X)

S2LS (O=UP, WEIGHT=3)

ERO2={A,C,X}

Note that we need to encode 2 END-POINTS objects here if we want to encode 2 S2LS objects, in order to conform to the RBNF of RFC 8623. If both EROs had the same weight (ECMP), then we would not need 2 S2LS objects and we would encode ERO2 directly after ERO1.
Use Case 2

Use Case 2: multiple optimization objectives sharing ECMP.

• Allocate a different PCEP Tunnel for every objective.
• Create a new PCEP association type (Ex., “ECMP Association”) to bind these Tunnels together.
Combination of both Use Cases

Going back to our previous example:
• LSP 1 and LSP 2 have different optimization objectives and ECMP is desired
• LSP 1 gets a single path X
• LSP 2 gets two ECMP paths: Y and Z
• As a result 50% of traffic goes on path X, 25% on path Y and 25% on path Z.

PCEP representation of the above:
• ECMP ASSOCIATION contains LSP 1 and LSP 2.
• LSP 1 contains a single ERO: X.
• LSP 2 contains two EROs: Y and Z.